LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
Il6r
no.698-702
cop. 2

[redacted]

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

NOV 5 1977

OCT 1 5 RECD

L161—O-1096

Il6r
no.700

Report No. UIUCDCS-R-75-700 NSF - OCA - GJ-36936 - 000008

A PARALLEL QR-ALGORITHM FOR TRIDIAGONAL SYMMETRIC MATRICES

by

Ahmed H. Sameh and David J. Kuck

February 1975

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

THE LIBRARY OF THE
MAR 24 1975
UNIVERSITY OF ILLINOIS

Digitized by the Internet Archive
in 2013

http://archive.org/details/parallelqralgori700same

A Parallel QR-Algorithm for Tridiagonal Symmetric Matrices*

by

Ahmed H. Sameh** and David J. Kuck***

*This work was supported in part by US NSF GJ-36936; and in part by the Advanced Research Project Agency of the Department of Defense under Contract No. DAHC04-72-C-0001.

**Department of Computer Science and the Center for Advanced Computation, University of Illinois, Urbana, Illinois 61801.

***Department of Computer Science
University of Illinois, Urbana, Illinois 61801.

Abstract

We show that if the size of the tridiagonal matrix in any given iteration is n, then the parallel QR-algorithm requires $O(\log_2 n)$ steps with $O(n)$ processors per iteration and no square roots. This results in a speedup of $O(n/\log_2 n)$ and an efficiency of $O(1/\log_2 n)$.

Keywords:

Givens reduction
Linear algebra
LU-decomposition
Parallelism

AMS(MOS) Subject Classification

Primary 68A20

Secondary 65F15

1. Introduction

Reinsch [1] proposed a stable QR algorithm for finding the eigenvalues of symmetric tridiagonal matrices that requires no square roots. In this paper we present a version of his algorithm that is suitable for parallel computers. We give upper bounds on the time and number of processors required, per iteration.

Throughout this paper we assume that any number of processors may be used at any time, but we will give bounds on this number. Each processor can perform any of the four arithmetic operations in one time step, and there are no memory or data alignment time penalties. We also give the following definitions. If $p \geq 1$ processors are used, we denote the computation time by T_p. The speedup of a parallel algorithm which uses p processors over a serial algorithm is defined by $S_p = T_1/T_p \geq 1$, and the corresponding efficiency of the computation is given by $E_p = S_p/p \leq 1$.

We show that if the size of the tridiagonal matrix in any given iteration is n, then the parallel QR algorithm requires $O(\log_2 n)$ steps per iteration (and no square roots) using $O(n)$ processors. This results in a speedup of $O(n/\log_2 n)$ and an efficiency of $O(1/\log_2 n)$. This algorithm depends on a parallel scheme developed by Chen and Kuck [2], for solving linear triangular systems.

2. The Algorithm

One iteration of the QR algorithm can be expressed in the form [3],

$$\begin{aligned} Q(A-kI) &= R \\ \tilde{A} &= RQ^t \end{aligned} \tag{1}$$

where k is an origin shift, Q is orthogonal, R is upper triangular, and we assume that the tridiagonal matrix A is of order $n = 2^m$, for a positive integer m. Note that the tridiagonal matrix $\tilde{A}$ is similar to (A-kI) rather than A. Let the diagonal elements of (A-kI) be denoted by a_i (i = 1, 2, ..., n) and the off-diagonal elements by b_i (i = 2, 3, ..., n). The orthogonal matrix Q is chosen as the product of (n-1) plane rotations, i.e., $Q = P_{n-1}^n \ldots P_2^3 P_1^2$, where P_j^{j+1} rotates rows j and j+1 annihilating the element in the position (j+1, j),

$$P_j^{j+1} = \left[\begin{array}{cccccccc} 1 &&&&&&& \\ & \ddots &&&&&& \\ && 1 &&&&& \\ &&& c_j & s_j &&& \\ &&& -s_j & c_j &&& \\ &&&&& 1 && \\ &&&&&& \ddots & \\ &&&&&&& 1 \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow \text{row } j \\ \\ \\ \\ \\ \end{array}$$

Therefore, if $P_{r-1}^{r} \ldots . P_2^3 P_1^2 (A-kI)$ is given by,

$$\begin{bmatrix} p_1 & q_2 & t_3 & & & \\ & p_2 & q_3 & t_4 & & \\ & \multicolumn{3}{c}{- - - - - - - - - -} & & \\ & & p_{r-1} & q_r & t_{r+1} & \\ & & & g_r & c_{r-1}\, b_{r+1} & \\ & & & b_{r+1} & a_{r+1} & b_{r+2} \\ & & & \multicolumn{3}{c}{- - - - - - - - - - - -} \end{bmatrix}$$

then it follows by induction that,

$$\begin{aligned} c_r &= g_r / p_r \\ s_r &= b_{r+1} / p_r \qquad\qquad r = 1, 2, \ldots, n - 1 \qquad\qquad (2) \\ q_{r+1} &= c_{r-1}\, c_r\, b_{r+1} + s_r\, a_{r+1} \end{aligned}$$

where,

$$g_r = c_{r-1}\, a_r - c_{r-2}\, s_{r-1}\, b_r \qquad r = 1, 2, \ldots, n \qquad\qquad (3)$$

in which $c_0 = 1$, $s_0 = 0$. From (2) and (3) we obtain the recurrence relation,

$$c_r\, p_r = c_{r-1}\, a_r - c_{r-2}\, s_{r-1}\, b_r \qquad r = 2, 3, \ldots, n - 1 \qquad\qquad (4)$$

Let $w_r = c_r \prod_{i=1}^{r} p_i$, then (4) reduces to the second order linear recurrence relation,

$$w_0 = 1, \; w_1 = \alpha_1$$

$$w_r = a_r \, w_{r-1} - b_r^2 \, w_{r-2} \qquad r = 2, 3, \ldots, n - 1$$

Solving (5) is equivalent to solving the lower triangular system of equations,

$$Lw = e_1$$

where,

$$w^t = (w_0, w_1, \ldots, w_{n-1}) \, ,$$

and $$\ell_{ij} = \begin{cases} 1 & i = j \\ -a_j & i = j + 1 \\ b_{j+1}^2 & i = j + 2 \\ 0 & \text{otherwise} \end{cases}$$

Using the identity $c_r^2 + s_r^2 = 1$, and defining $z_r^2 = \prod_{i=1}^{r} p_i^2$, $r = 1, 2, \ldots,$ $n - 1$, we obtain the first order linear recurrence relation

$$z_0^2 = w_0^2 = 1$$

$$z_r^2 = b_{r+1}^2 \, z_{r-1}^2 + w_r^2 \qquad r = 1, 2, \ldots, n - 1$$

This is equivalent to the lower triangular system,

$$\tilde{L}\tilde{z} = \tilde{w}$$

where,

$$\tilde{w}^t = (w_0^2, w_1^2, \ldots, w_{n-1}^2) ,$$

$$\tilde{z}^t = (z_0^2, z_1^2, \ldots, z_{n-1}^2) ,$$

and $$\tilde{\ell}_{ij} = \begin{cases} 1 & i = j \\ -b_i^2 & i = j + 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that the squares of the subdiagonal elements rather than the elements themselves are the given data. So squaring these elements of the original matrix will cost only one step and (n-1) processors, which is negligible compared to the time of finding all the eigenvalues of the tridiagonal matrix.

The linear systems (6) and (8) are solved sequentially. Chen and Kuck [2] have shown that any m-th order linear recurrence system of n equations, or any unit lower triangular system of bandwidth m + 1, can be solved in

$$T_p = (2+\log_2 m)\log_2 n - \frac{1}{2}(\log_2^2 m+\log_2 m)$$

steps with $p = O(m^2 n)$ processors. Specifically, the system (6) can be solved in $(3\log_2 n - 1)$ steps with no more than 4n processors. Once w is obtained, forming the right-hand side $\tilde{w}$ in (8) requires one step with (n-1) processors, and the system itself can be solved in $2\log_2 n$ steps with (n-1) processors. Thus, solving (6) and (8) requires $T_{p_1} = 5\log_2 n$ steps with $p_1 = 4n$ processors. From w and $\tilde{z}$ we can obtain,

$$p_j^2 = z_j^2/z_{j-1}^2 \,,$$

$$c_j^2 = w_j^2/z_j^2 \,, \qquad j = 1, 2, \ldots, n - 1 \qquad ($$

$$s_j^2 = b_{j+1}^2/p_j^2 \,,$$

and, $$p_n^2 = \eta^2/z_{n-1}^2$$

in which $\eta = (a_n \, w_{n-1} - b_n^2 \, w_{n-2})$ and where we have used (3) and (9).

It can be shown [4] that the orthogonal matrix Q is lower Hessenberg in which $Q_{jj} = c_{j-1} \, c_j = w_{j-1} \, w_j/z_{j-1} \, z_j$ $(j = 1, 2, \ldots, n - 1$ $Q_{nn} = c_{n-1}$, and $Q_{j,j+1} = s_j$. Therefore, the elements of the tridiagonal matrix $\tilde{A} = RQ^t$ are given by,

$$\tilde{a}_j = \xi_j \, \tau_j + a_{j+1} \, s_j^2 \,,$$

$$\tilde{b}_{j+1}^2 = p_{j+1}^2 \, s_j^2 \,, \qquad j = 1, 2, \ldots, n - 1 \qquad ()$$

and $$\tilde{a}_n = \xi_n \, a_n - \xi_{n-1} \, b_n^2$$

where,

$$\tau_j = (b_{j+1}^2 + p_j^2) \,,$$

$$\xi_j = w_{j-1} \, w_j/z_j^2 \,, \qquad j = 1, 2, \ldots, n - 1 \qquad ()$$

and $$\xi_n = w_{n-1}^2/z_{n-1}^2$$

We show in Figure 1 that the elements of $\tilde{A}$ can be evaluated in $T_{p_2} = 4$ steps using $p_2 = 3n$ processors. Thus, the total time required is given by $T_p = T_{p_1} + T_{p_2} = 4 + 5\log_2 n$ steps with $p = 4n$ processors. Reinsch [1] has shown that on a serial computer the time required for one iteration is $T_1 = 11n$ steps. Hence, our parallel algorithm yields a speedup $S_p \simeq 11n/5\log_2 n$ with an efficiency $E_p \simeq 11/20\log_2 n$.

The linear systems (6) and (8) can be extremely ill-conditioned. For example, for Wilkinson's matrix W_{21}^{-} [5], $a_i = 11 - i$ $(i = 1, 2, \ldots, 21)$ and $b_i = 1$, the condition number of L, in (6), in the first iteration is of the order 10^{-17}. We simulated the above algorithm on an IBM 360/75, and in spite of such high condition number we obtained all the eigenvalues correct to at least 9 decimal places. Wilkinson [6] has shown that the computed solution of a triangular system has low relative errors. Specifically, system (8), $(\tilde{w}_i > 0,\ \tilde{\ell}_{ii} > 0,\ \tilde{\ell}_{ij} \leq 0\ i \neq j)$, will have low relative error in the solution no matter how ill-conditioned $\tilde{L}$ may be.

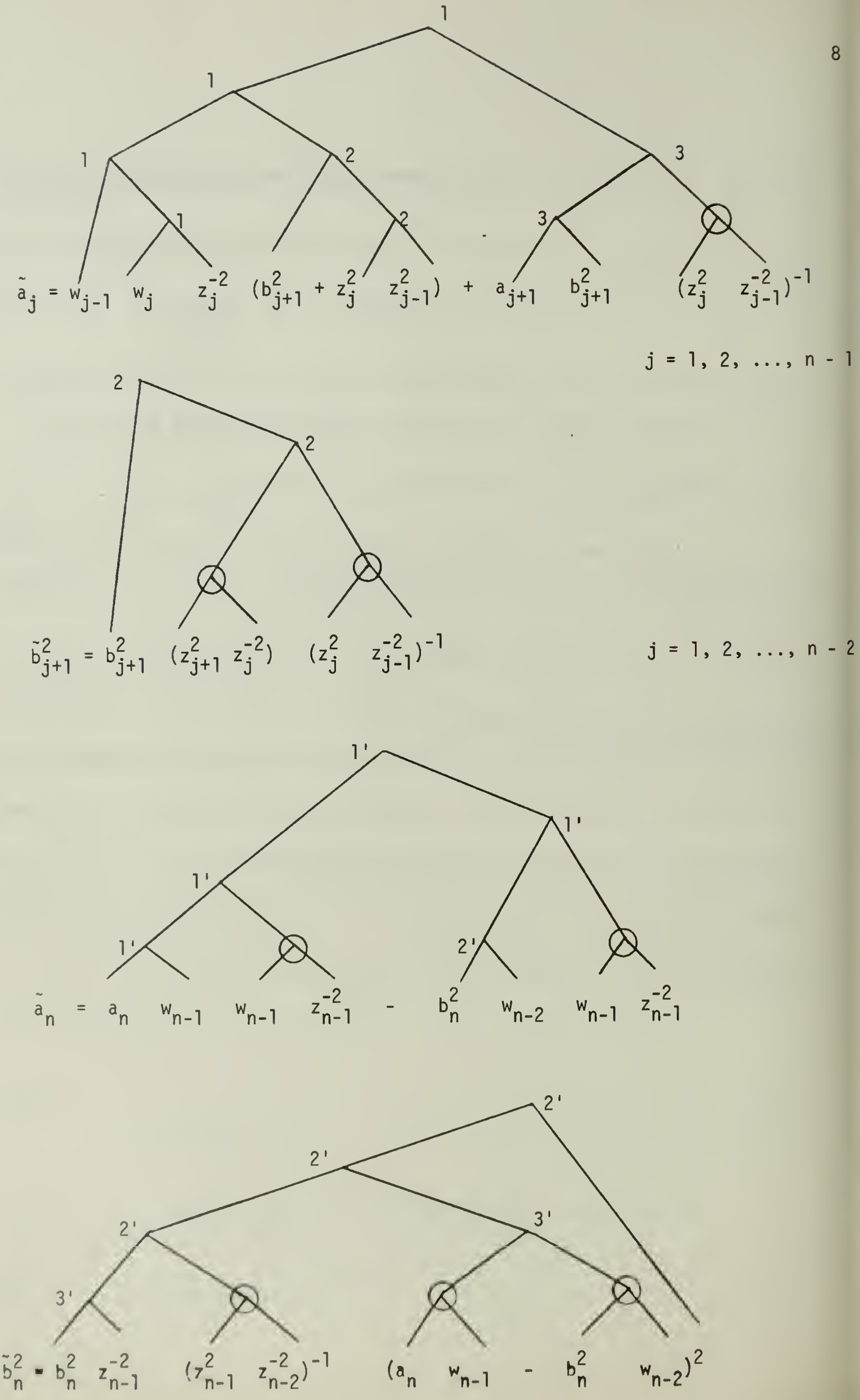
1
1
1
2
3
1
2
3
$\tilde{a}_j = w_{j-1} \; w_j \; z_j^{-2} \; (b_{j+1}^2 + z_j^2 \; z_{j-1}^2) + a_{j+1} \; b_{j+1}^2 \; (z_j^2 \; z_{j-1}^{-2})^{-1}$
$j = 1, 2, \ldots, n - 1$
2
2
$\tilde{b}_{j+1}^2 = b_{j+1}^2 \; (z_{j+1}^2 \; z_j^{-2}) \; (z_j^2 \; z_{j-1}^{-2})^{-1}$
$j = 1, 2, \ldots, n - 2$
1'
1'
1'
1'
2'
$\tilde{a}_n = a_n \; w_{n-1} \; w_{n-1} \; z_{n-1}^{-2} - b_n^2 \; w_{n-2} \; w_{n-1} \; z_{n-1}^{-2}$
2'
2'
2'
3'
3'
$\tilde{b}_n^2 = b_n^2 \; z_{n-1}^{-2} \; (z_{n-1}^2 \; z_{n-2}^{-2})^{-1} \; (a_n \; w_{n-1} - b_n^2 \; w_{n-2})^2$

Figure 1

REFERENCES

[1] C. H. Reinsch, "A stable, rational QR algorithm for the computation of the eigenvalues of an Hermitian, tridiagonal matrix," Math. Comp., v. 25, 1971, pp. 591-597. MR 45 # 4621.

[2] S. C. Chen, and D. J. Kuck, "Time and parallel processor bounds for linear recurrence systems," to appear IEEE Transactions on Computers.

[3] H. Bowdler, R. S. Martin, C. Reinsch, and J. H. Wilkinson, "The QR and QL algorithms for symmetric matrices," Numer. Math., v. 11, 1968, pp. 293-306.

[4] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," Math. Comp., v. 28, 1974, pp. 505-535.

[5] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965. MR 32 # 1894.

[6] J. H. Wilkinson, "Error analysis of direct methods of matrix inversion," J. ACM, v. 8, 1961, pp. 281-330.

OGRAPHIC DATA

1. Report No. UIUCDCS-R-75-700

2.

3. Recipient's Accession No.

Subtitle

PALLEL QR-ALGORITHM FOR TRIDIAGONAL SYMMETRIC MATRICES

5. Report Date February 1975

6.

ed H. Sameh and David J. Kuck

8. Performing Organization Rept. No. UIUCDCS-R-75-700

Organization Name and Address

iversity of Illinois at Urbana-Champaign
partment of Computer Science
ana, IL 61801

10. Project Task Work Unit No.

11. Contract Grant No.

US NSF-GJ-36936

Organization Name and Address

ional Science Foundation
hington, D. C. 20550

13. Type of Report & Period Covered

Technical Report

14.

We show that if the size of the tridiagonal matrix in any given iteration is n, then the parallel QR-algorithm requires $O(\log_2 n)$ steps with $O(n)$ processors iteration and no square roots. This results in a speedup of $O(n/\log_2 n)$ and an ficiency of $O(1/\log_2 n)$.

nd Document Analysis. 17a. Descriptors

ens reduction
ear algebra
decomposition
allelism

Open-Ended Terms

ld/Group

Statement

mited

19. Security Class (This Report) UNCLASSIFIED

20. Security Class (This Page) UNCLASSIFIED

21. No. of Pages 12

22. Price

[illegible]

BOUND BY
HNM

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 no.698-702(1975
Report /
3 0112 088401747